5 渲染流水线

5.1 3D视觉即错觉

怎么样将3D场景的空间感和立体感在2D平面显示器的屏幕上表现出来呢？

1. 近大远小；
2. 光照和阴影；

5.2 模型的表示

实体3D对象是借助三角形网格来近视表示的，三角形是3D物体建模的基石。

除了三角形，点和面也有用武之地。

5.3 计算机色彩基础

5.3.1 颜色运算

颜色向量也有它自己专属的颜色运算，即分量式乘法。

5.3.2 128位颜色

DirectXMath库针对分量式运算提供了函数：XMColorModulate。

5.3.3 32位颜色

一般来说，128位颜色值常用于高精度的颜色运算(例如位于像素着色器中的各种运算)。但是，最终存储在后台缓冲区中的像素颜色数据，却往往都是以32位颜色值来表示。

5.4 渲染流水线概述

渲染流水线是以摄像机为观察视角生成2D图像的一系列完整步骤。

渲染流水线：

输入装配器阶段

顶点着色器阶段

外壳着色器阶段

曲面细分阶段

域着色器阶段

几何着色器阶段

光栅化阶段

像素着色器阶段

输出合并阶段

5.5 输入装配器阶段

输入装配器阶段会从显存中读取几何数据，再将它们装配为几何图元。简单来说，我们是通过索引来定义如何将定点装配在一起，从而构成图元的方法。

5.5.1 顶点

5.5.2 图元拓扑

在Direct3D中，我们要通过一种名为顶点缓冲区的特殊数据结构，将定点与渲染流水线相绑定。顶点缓冲区利用连续的内存来存储一系列顶点。可是，仅凭这一点并不能说明这些顶点究竟如何组成几何图元。对此，我们要通过指定图元拓扑来告知Direct3D如何用顶点数据来表示几何图元。

5.5.2.1 点列表

所有顶点都被绘制为一个单独的点。

5.5.2.2 线条带

顶点被连接成一系列的连续线段。有n+1个顶点就会生成n条线段。

5.5.2.3 线列表

每对顶点都会组成单独的线段，2n个顶点就会生成n条线段。

5.5.2.4 三角形带

三角形连接成带状，处于中间的顶点被相邻的三角形所共用。因此，利用n个顶点即可生成n-2个三角形。

5.5.2.5 三角形列表

将每3个顶点装配成独立的三角形，所以每3n个顶点会生成n个三角形。

5.5.2.6 具有邻接数据的图元拓扑

对于存有邻接数据的三角形列表而言，每个三角形都有3个与之相邻的邻接三角形。

5.5.2.7 控制点面片列表

将顶点数据解释为具有N个控制点的面片列表。此图元常用于渲染流水线的曲面细分阶段。

5.5.3 索引

为三角形指定顶点顺序是一项十分重要的工作，我们称这个顺序为绕序。

整个工作流程是这样的：先创建一个顶点列表和一个索引列表。在顶点列表中收录一份所有独立的顶点，并在索引列表中存储顶点列表的索引值。这些索引定义了顶点列表中的顶点是如何组合在一起，从而构成三角形的。

5.6 顶点着色器阶段

我们可以把顶点着色器看作一种输入与输出数据皆为单个顶点的函数。

5.6.1 局部空间和世界空间

局部坐标系通常是一种以目标物体的中心为原点，并且坐标轴与该物体对齐的简便易用坐标系。

5.6.2 观察空间

在此坐标系中，该虚拟摄像机位于原点并沿z轴的正方向观察，x轴指向摄像机的右侧，y轴只想摄像机的上方。

只要给定摄像机的位置、观察目标点以及世界空间中“向上”方向的向量，我们就能构建出对应的摄像机局部坐标系，并推导出相应的观察矩阵。

5.6.3 投影和齐次裁剪空间

摄像机可观察的空间体积，可用一个由四棱锥截取的平截头体来表示。

下一个任务是将平截头体内的3D几何体投影到一个2D投影窗口中。

5.6.3.1 定义平截头体

在观察空间中，我们可以通过近平面n、远平面f、垂直视场角α以及横纵比r这4个参数来定义一个：以原点作为投影的中心，并沿z轴正方向进行观察的平截头体。

由于图像终将被映射到后台缓冲区中，因此，我们希望令投影窗口与后台缓冲区两者的横纵比保持一致。为此，我们通常将投影窗口的横纵比指定为后台缓冲区的横纵比。

一旦给定垂直视场角α和横纵比r，我们必能求出水平视场角β：

β = 2arctan(r \* tan(α / 2))

5.6.3.2 投影顶点

我们希望求出给定点(x,y,z)在投影屏幕z=d中的投影(x’,y’,d),通过在x轴和y轴上分别利用相似三角形性质可求出。

5.6.3.3 规格化设备坐标

如果能去除投影窗口对横纵比的依赖，那么处理过程会更加简单，为此，我们将x坐标上的投影区间从[-r,r]缩放至归一化区间[-1,1]。

经此映射处理后，x坐标和y坐标就成为了规格化设备坐标(NDC)。

在NDC坐标中，投影窗口的高和宽都是2。

5.6.3.4 用矩阵来表示投影公式

除以w的计算过程有时被称为透视除法或齐次除法。

5.6.3.5 归一化深度值

g(z)大部分取值是由近平面附近的深度值计算得出的。大多数的深度值被集中地映射到了取值区间中的一段较小的区域内。这将引发深度缓冲区的精度问题，对此，我们一般建议令近平面与远平面尽可能地接近，以改善深度值的精度问题。

透视投影矩阵

在顶点乘以投影矩阵之后但还未进行透视除法之前，几何体会处于所谓的齐次裁剪空间。待完成透视除法之后，便是用规格化设备坐标(NDC)来表示几何体了。

5.6.3.6 XMMatrixPerspectiveFovLH函数

利用该函数来构建透视投影矩阵。

5.7 曲面细分阶段

曲面细分阶段是利用镶嵌化处理技术对网格中的三角形进行细分，以此来增加物体表面的三角形数量。再将这些新增的三角形偏移到适当的位置，使网格表现出更加细腻的细节。

5.8 几何着色器阶段

几何着色器接受的输入应当是完整的图元。几何着色器的主要优点是可以创建或销毁几何体。几何着色器的常见拿手好戏是将一个点或一条线扩展为一个四边形。

5.9 裁剪

处于平截头体交界处的几何体将会被裁剪。裁剪操作是由硬件负责的。

5.10 光栅化阶段

光栅化阶段的主要任务是为投影至屏幕上的3D三角形计算出对应的像素颜色。

5.10.1 视口变换

当裁剪操作完成之后，硬件会通过透视除法将物体从齐次裁剪空间变换为规格化设备坐标(NDC)。一旦物体的顶点位于NDC空间内，构成2D图像的2D顶点x、y坐标就会被变换到后台缓冲区中的视口矩形里。

5.10.2 背面剔除

每个三角形都有两个面。

根据观察者的视角看去，顶点绕序为顺时针方向的三角形为正面朝向，而顶点绕序为逆时针方向的三角形为背面朝向。

由于背面朝向的三角形都被正面朝向的三角形所遮挡，所以绘制它们是没有意义的。背面剔除就是用于将背面朝向的三角形从渲染流水线中除去的处理流程。

5.10.3 顶点属性插值

为了得到屏幕空间中各个顶点的插值属性，往往要通过一种名为透视矫正插值的方法，对3D空间中的三角形属性进行线性插值。

我们无须考虑透视矫正插值法处理像素属性的具体数学细节，因为硬件会自动完成相应的处理。

5.11 像素着色器阶段

5.12 输出合并(OM)阶段

在此阶段中，一些像素片段可能会被丢弃。

混合操作也是此阶段实现的。